# Introduction

## TracNav

- JPFWiki - Welcome Page
- **Introduction**
  - What is JPF
  - Testing vs model checking
  - Random Example
  - Race Example
  - JPF classification
- **Installing JPF...**
- **User Guide...**
- **Developer Guide...**
- **Projects...**
- Summer Projects
- External Projects
- Change(B)log
- **About...**
- Events
- Presentations
- Papers
- FAQ
- History?
- Support
- People?
- Playground
- Table of Context

You might have noticed that we were a bit vague about what JPF actually is. It started as a software model checker, but nowadays there are various different execution modes and extensions, hence we have to explain what JPF does without getting lost in details of its application. What all JPF modes, which are runtime configured and not hardwired, have in common is that they are used to verify Java programs, to

- find and explain defects
- collect "deep" runtime information like coverage metrics
- deduce interesting test vectors and create corresponding test drivers
- and many more...

Although classic SW model checking is only one of these applications, it is still what JPF is mostly associated with. People often confuse this with testing, and indeed JPFs notion of model checking can be close to systematic testing, so we throw in a little example that illustrates the differences.

Here is the outline of this section:

- What is JPF?
- Testing vs. Model Checking
  - Random value example

    - ♦ Data race example
- JPF key features